

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 182

**INKREMENTALNO UČENJE MODELA ZA  
RASPOZNAVANJE SLIKE UZ DODAVANJE NOVIH  
RAZREDA**

Dario Deković

Zagreb, lipanj 2021.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 182

**INKREMENTALNO UČENJE MODELA ZA  
RASPOZNAVANJE SLIKE UZ DODAVANJE NOVIH  
RAZREDA**

Dario Deković

Zagreb, lipanj 2021.

## ZAVRŠNI ZADATAK br. 182

Pristupnik: **Dario Deković (0036515892)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: prof. dr. sc. Siniša Šegvić

Zadatak: **Inkrementalno učenje modela za raspoznavanje slike uz dodavanje novih razreda**

### Opis zadatka:

Raspoznavanje slika važan je zadatak računalnog vida s mnogim zanimljivim primjenama. Trenutno stanje tehnike koristi duboke modele koji se uče s kraja na kraj. Međutim, tako dobivene modele nije lako prilagođavati novim zadacima zbog katastrofalnog zaboravljanja. Zbog toga inkrementalno učenje novih razreda predstavlja zanimljiv izazov. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće pristupe za klasifikaciju slike. Odabrati slobodno dostupni skup slika te oblikovati podskupove za učenje, validaciju i testiranje. Predložiti prikladnu arhitekturu dubokog klasifikacijskog modela. Uhodati postupke inkrementalnog učenja te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 11. lipnja 2021.

*Zahvaljujem svom mentoru prof. dr. sc Siniši Šegviću na pomoći prilikom izrade završnog rada.*

*Hvala mojoj sestri na pomoći i savjetima.*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Duboko učenje</b>	<b>2</b>
2.1. Umjetne neuronske mreže . . . . .	2
2.1.1. Umjetni neuron . . . . .	3
2.1.2. Unaprijedna neuronska mreža . . . . .	5
2.2. Duboke neuronske mreže . . . . .	8
2.3. Konvolucijske neuronske mreže . . . . .	8
2.3.1. Uvod . . . . .	8
2.3.2. Matematička konvolucija . . . . .	8
2.3.3. Konvolucijski sloj . . . . .	9
2.3.4. Sloj sažimanja . . . . .	10
<b>3. Inkrementalno učenje uz dodavanje novih razreda</b>	<b>11</b>
3.1. Uvod . . . . .	11
3.2. Podjela pristupa . . . . .	12
3.2.1. Algoritmi bazirani na rastu modela . . . . .	12
3.2.2. Algoritmi bazirani na fiksiranom prikazu . . . . .	12
3.2.3. Algoritmi bazirani na finom podešavanju . . . . .	13
<b>4. Konsolidacija dubokih modela</b>	<b>14</b>
4.1. Uvod . . . . .	14
4.2. Faza treniranja novog modela . . . . .	15
4.3. Faza konsolidacije . . . . .	15
4.4. Faza finog podešavanja . . . . .	17
<b>5. Implementacija</b>	<b>18</b>
5.1. Okvir za automatsko diferenciranje . . . . .	18
5.2. Skupovi podataka . . . . .	18

5.2.1.	CIFAR-100 . . . . .	18
5.2.2.	ImageNet32 . . . . .	18
5.2.3.	Uvećanje podataka . . . . .	19
5.3.	Arhitektura modela . . . . .	19
5.3.1.	Rezidualne mreže . . . . .	19
5.4.	Hiperparametri i detalji treniranja . . . . .	20
<b>6.</b>	<b>Eksperimenti</b>	<b>21</b>
6.1.	Konfiguracija 50/50 . . . . .	22
6.2.	Konfiguracija 20/20 . . . . .	24
6.3.	Konfiguracija 10/10 . . . . .	26
6.4.	Konfiguracija 5/5 . . . . .	28
6.5.	Rasprava . . . . .	30
<b>7.</b>	<b>Zaključak</b>	<b>31</b>
	<b>Literatura</b>	<b>32</b>

# 1. Uvod

Umjetna inteligencija jedna je od glavnih pokretačkih sila modernog svijeta. Grana umjetne inteligencije koja u zadnjih nekoliko godina doživljava najveći procvat je duboko učenje. Duboko učenje jednu od svojih najbitnijih primjena nalazi u području računalnog vida. Zadaci koje postavljamo pred algoritme dubokog učenja u računalnom vidu različiti su, a jedan od najstarijih je raspoznavanje slika. Većina modernih algoritama dubokog učenja osmišljena je za učenje kada su unaprijed poznati svi razredi koje će biti potrebno raspoznati. Takav uvjet pokazuje se presnažnim za primjenu u sustavima za rad u stvarnom vremenu te je pokušaj rješavanja tog problema doveo do nove grane dubokog učenja, inkrementalno učenje uz dodavanje novih razreda.

Inkrementalno učenje uz dodavanje novih razreda razmatra realan scenarij u kojem nisu unaprijed poznate sve klase koje model mora raspoznati. Jedna od novijih metoda inkrementalnog učenja uz dodavanje novih razreda je konsolidacija dubokih modela. Konsolidacija dubokih modela, kao i ostali algoritmi inkrementalnog učenja, temelji se na konvolucijskim neuronskim mrežama. Unatoč tome, za same metode, vrsta mreže najčešće nije presudna te se tome u radovima o inkrementalnom učenju pridaje manje pažnje.

U ovom radu opisat ću teorijsku podlogu dubokog učenja. Detaljnije ću se dotaknuti konvolucijskih neuronskih mreža i njihovih gradivnih elemenata. Definirat ću inkrementalno učenje uz dodavanje novih razreda te opisati podjelu algoritama unutar inkrementalnog učenja. Zatim ću detaljno opisati i implementirati konsolidaciju dubokih modela. Provesti ću ispitivanje dobivene implementacije na skupu podataka CIFAR-100 za različite konfiguracije problema inkrementalnog učenja. Za kraj ću se osvrnuti na dobivene rezultate i usporediti konfiguracije.

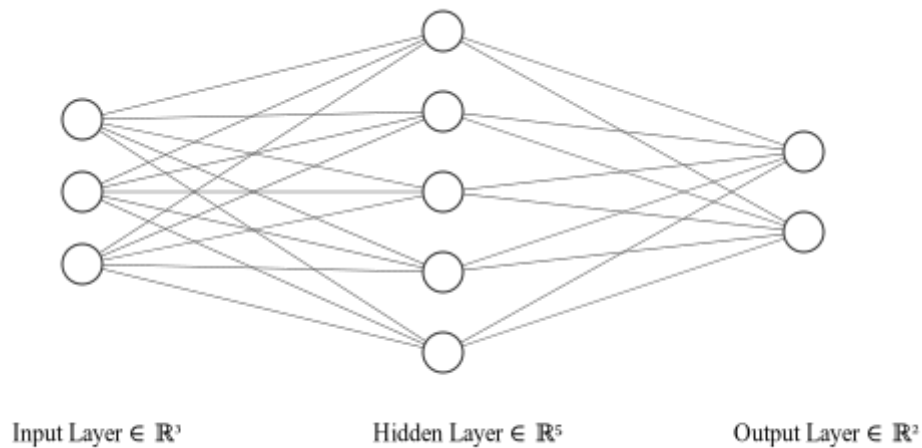
## 2. Duboko učenje

Umjetne neuronske mreže (engl. *Artificial neural network*) jedan su od glavnih alata u području strojnog učenja. Svoje začetke imaju u 40-im godinama prošlog stoljeća kada su Warren McCulloch i Walter Pitts modelirali jednostavnu neuronsku mrežu koristeći električne krugove [17]. Daljnji razvoj u području obilježilo je nekoliko ključnih događaja koji su odredili dijelove današnjih umjetnih neuronskih mreža i te ćemo dijelove pobliže proučiti u narednom poglavlju. Prvi od tih događaja bio je razvoj perceptrona od strane Franka Rosenblatta 1958. godine. Perceptron je sustav izgrađen na temelju McCulloch-Pitts modela umjetnog neurona te je izazvao veliki optimizam u AI zajednici. Unatoč tome, perceptron je imao nekoliko nedostataka koji su istaknuti u knjizi "Perceptrons" koja je izdana 1969. godine. Svojom knjigom Marvin Minsky i Seymour Papert skrenuli su smjer istraživanja u umjetnoj inteligenciji s konektivističkog prema simboličkom pristupu te uveli područje u period koji se danas popularno naziva "zima umjetne inteligencije (engl. *AI winter*). Početkom 80-ih godina neuronske mreže doživljavaju svoju renesansu zahvaljujući algoritmu unazadne propagacije pogreške. Zbog hardverskih ograničenja neuronske mreže doživjele su ponovni pad u popularnosti početkom ovog stoljeća. Ipak, razvojem grafičkih kartica za potrebe industrije elektroničkih igara, omogućena je izgradnja neuronskih mreža od velikog broja slojeva što nas je uvelo u novu eru, eru dubokog učenja.

### 2.1. Umjetne neuronske mreže

Umjetne neuronske mreže temelj su konektivističkog pristupa umjetnoj inteligenciji. Sastoje se od više međusobno povezanih slojeva koji se sastoje od jednog ili više umjetnih neurona. Slojevi se dijele na ulazni sloj, izlazni sloj i jedan ili više skrivenih slojeva. Na slici 2.1 prikazana je arhitektura potpuno povezane mreže s jednim skrivenim slojem.



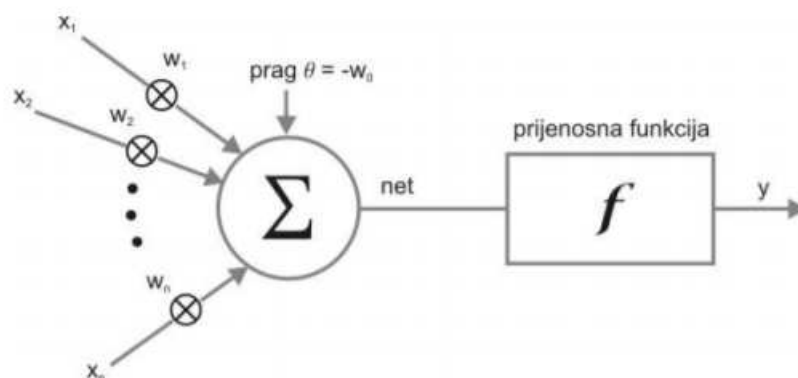


**Slika 2.1:** Model umjetne neuronske mreže.

Kao i sve metode strojnog učenja, umjetne neuronske mreže sastoje se od tri glavne komponente modela, funkcije gubitka i optimizacijskog postupka. Ove tri komponente, uz umjetni neuron, čine temelj umjetnih neuronskih mreža te ću ostatak odjeljka posvetiti njihovom detaljnije opisivanju.

### 2.1.1. Umjetni neuron

Umjetni neuron gradivna je jedinica neuronskih mreža. Komponente koje sačinjavaju umjetni neuron su ulazni signali pomnoženi s težinskim faktorom i zbrojeni te aktivacijska funkcija kroz koju se propušta dobiveni zbroj [1]. Izlaz iz aktivacijske funkcije je ujedno i izlazna vrijednost umjetnog neurona kao što je vidljivo na slici 2.2.



**Slika 2.2:** Model umjetnog neurona. Slika je preuzeta iz [3].

Postoje različite vrste umjetnih neurona koje su jednoznačno određene korištenom

aktivacijskom funkcijom. U nastavku dan je pregled najpopularnijih.

- **Linearna funkcija**

$$f(net) = net \quad (2.1)$$

Linearna funkcija najjednostavnija je aktivacijska funkcija. Kao što je vidljivo iz jednadžbe 2.1 ta funkcija kao izlaz vraća upravo zbroj svih ulaza pomnoženih težinskim faktorom. Model umjetnog neurona koji koristi ovu funkciju naziva se ADALINE (engl. *Adaptive Linear Element*) te je jedan od prvih modela umjetnog neurona u povijesti razvijen na Stanfordu davne 1959. godine [17].

- **Funkcija praga**

$$f(net) = \begin{cases} 0 & \text{za } net < 0 \\ 1 & \text{inače} \end{cases} \quad (2.2)$$

Funkcija praga jedna je od najstarijih aktivacijskih funkcija, korištena je još u Rosenblattovom perceptronu. TLU (engl. *Threshold Logic Unit*) perceptron naziv je umjetnog neurona koji koristi ovu aktivacijsku funkciju. Porijeklo naziva možemo iščitati iz jednadžbe 2.2. Naime, ova aktivacijska funkcija daje Booleov izlaz.

- **Sigmoidalna funkcija**

$$f(net) = \frac{1}{1 + e^{-a*net}} \quad (2.3)$$

Povijesno najčešće korištena aktivacijska funkcija, sigmoidalna funkcija može se smatrati kao izgladnena verzija funkcije praga. Glavna prednost sigmoidalne funkcije u odnosu na funkciju praga je to što je beskonačno mnogo puta diferencijabilna na cijeloj svojoj domeni, dok funkcija praga nije diferencijabilna u nuli. S druge strane, sigmoidalna funkcija podložna je problemu nestajućih gradijenata te se danas rijetko koristi u modernim dubokim neuronskim mrežama [10].

- **ReLU funkcija**

$$f(net) = \begin{cases} net & \text{za } net > 0 \\ 0 & \text{inače} \end{cases} \quad (2.4)$$

ReLU (engl. *Rectified Linear Unit*) funkcija danas je najpopularnija aktivacijska

funkcija. Svoju popularnost može zahvaliti svojoj primjeni u dubokim neuronskim mrežama. Kao što je vidljivo iz jednadžbe 2.4 ReLU propušta pozitivne izlaze dok negativne izlaze pretvara u nulu. Zbog velikog dijela svoje domene gdje je korisno diferencijabilna (prva derivacija je veća od nule) ReLU funkcija nije podložna problemu nestajućeg gradijenta. Nadalje, ReLU funkcija i njene derivacije su izrazito lagane za izračunati i implementirati. [10]. Iz tih razloga, ReLU funkcija je preporučena funkcija za korištenje u modernim dubokim neuronskim mrežama [6].

### 2.1.2. Unaprijedna neuronska mreža

Unaprijedna neuronska mreža (engl. *feedforward neural network*) temeljni je model neuronskih mreža. Naziv unaprijedna potječe iz smjera toka informacija koje se kroz mrežu kreću isključivo od ulaznog prema izlaznom sloju. Cilj takve neuronske mreže aproksimacija je neke funkcije  $f$ . Unaprijedna neuronska mreža definira mapiranje  $y = \hat{f}(x; \theta)$  te uči vektor parametara  $\theta$  koji rezultira najboljom aproksimacijom funkcije  $f$  [6]. Vektor parametara  $\theta$  je skup, prije spomenutih, težinskih faktora kojima množimo ulaze u umjetni neuron.  $\hat{f}$  predstavlja prije spomenuti model u okviru strojnog učenja.

Nakon modela potrebno je definirati sljedeću važnu komponentu neuronskih mreža, funkciju gubitka. Cilj treniranja neuronske mreže minimizacija je funkcije gubitka. Općenito, funkciju gubitka definiramo kao u jednadžbi 2.5 gdje je  $y$  stvarna vrijednost za neki ulaz  $x$ , a  $\hat{y}$  predstavlja izlaz iz neuronske mreže za dani ulaz  $x$ .

$$\begin{aligned} J(\theta) &= \text{cost}(y, \hat{y}) \\ \hat{y} &= \hat{f}(x; \theta) \end{aligned} \tag{2.5}$$

Dvije specifične funkcije gubitka koje sam koristio u ovom radu su kvadratni gubitak i gubitak unakrsne entropije.

- **Kvadratni gubitak**

$$J = \frac{1}{t} \sum_{j=1}^t (y^j - \hat{y}^j)^2 \tag{2.6}$$

Kao što je vidljivo iz formule 2.6 kvadratni gubitak sastoji se od usrednjene kvadrirane razlike očekivanog i dobivenog izlaza iz modela. Često se koristi prilikom prijenosa znanja s jedne mreže na drugu.

- **Gubitak unakrsne entropije**

$$J = \sum_{i=1}^N y_i \log f(\hat{y}_i) \quad (2.7)$$

$$f(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_{i=1}^N e^{\hat{y}_i}} \quad (2.8)$$

Gubitak unakrsne entropije često se koristi u problemima raspoznavanja slika i klasifikaciji, općenito. U formuli 2.7 vidimo kako ga definiramo kao negativni logaritam izlaza jednog neurona zadnjeg sloja pomnožen sa očekivanim izlazom tog neurona. Zatim taj rezultat zbrajamo po svim mogućim razredima gdje u formuli 2.7  $N$  predstavlja broj razreda. Kako bi izlazi bili korisni za računanje gubitka unakrsne entropije koristimo softmax funkciju koja je opisana u formuli 2.8. Softmax funkcija može se interpretirati kao vjerojatnost razreda.

Kao što je već spomenuto cilj treniranja neuronske mreže je  $\min(J(\theta))$ . Taj cilj postizemo koristeći se prije spomenutim postupkom optimizacije kako bi aproksimacijom došli do najbolje vrijednosti parametra  $\theta$ .

Gradijentni spust jedan je od najpopularnijih optimizacijskih algoritama i najčešći način za optimizaciju neuronskih mreža [15]. Njegovu popularnost možemo objasniti time što druge metode optimizacije ne funkcioniraju kada model ima više milijuna parametara što je čest slučaj u dubokim neuronskim mrežama. Gradijentni spust predstavlja način minimizacije prije opisane funkcije gubitka  $J(\theta)$  kroz ažuriranje parametra  $\theta$  u smjeru suprotnom od gradijenta funkcije gubitka. Veličina koraka kojim se krećemo određena je korakom učenja  $\eta$ . Gradijentni spust može se podijeliti u tri skupine koje se razlikuju po količini podataka koju koristimo za izračunavanje gradijenta.

- **Gradijentni spust**

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta) \quad (2.9)$$

Gradijentni spust grupe provodi ažuriranje parametra  $\theta$  s gradijentom funkcije gubitka izračunatim nad cijelim skupom podataka. Iz tog razloga gradijentni spust grupe može biti jako spor. Nije ga moguće koristiti za skupove podataka koji ne stanu cijeli u memoriju. S druge strane, gradijentni spust garantirano konvergira u globalni minimum za funkcije gubitka s konveksnom površinom ili lokalni minimum za nekonveksne funkcije.

- **Stohastički gradijentni spust**

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (2.10)$$

Nasuprot gradijentnom spustu grupe imamo stohastički gradijentni spust. Kao što vidimo iz jednadžbe 2.10 stohastički gradijentni spust ažurira parametar  $\theta$  za svaki ulaz  $x^{(i)}$  i pripadajući ispravni izlaz  $y^{(i)}$ . Zbog toga je stohastički gradijentni spust vrlo brz, ali zbog čestih ažuriranja ima visoku varijancu te, samim time, konvergencija može izostati.

- **Gradijentni spust preko mini-grupa**

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \quad (2.11)$$

Gradijentni spust mini grupe predstavlja najbolje od obje prije opisane verzije gradijentnog spusta. Ažuriranja parametra  $\theta$  provode se nakon izračunavanja gradijenta funkcije gubitka za grupu veličine  $n$  ulaza  $x^{(i:i+n)}$  i pripadajućih ispravnih izlaza  $y^{(i:i+n)}$ . Kombinacija brzog izračunavanja gradijenata i smanjene varijance funkcije gubitka čini gradijentni spust mini grupe najpopularnijim izborom za treniranje neuronskih mreža.

Uz gradijentni spust i njegove varijacije u ovom odlomku smo uveli i dva hiperparametra koja je potrebno izabrati prilikom treniranja neuronske mreže. Ti hiperparametri su korak učenja  $\eta$  i veličina mini grupe  $n$  i o njima će više biti rečeno u poglavlju posvećenom implementaciji.

Računanje gradijenata funkcije gubitka spomenuto je kroz definiranje gradijentnog spusta bez spominjanja konkretnog rješenja za učinkovitu provedbu tog računa. U tu svrhu ćemo koristiti jedan od najvažnijih algoritama u području umjetnih neuronskih mreža, algoritam unazadne propagacije pogreške. Algoritam unazadne propagacije pogreške omogućava izračunavanje gradijenta funkcije gubitka za svaki težinski faktor u mreži. To nadalje omogućava ažuriranje svakog težinskog faktora posebno u svrhu postepenog smanjivanja funkcije gubitka kroz više iteracija treniranja [19].

## 2.2. Duboke neuronske mreže

Duboke neuronske mreže čine okosnicu modernog pristupa umjetnoj inteligenciji. Općenito, duboke neuronske mreže su umjetne neuronske mreže s više slojeva između ulaznog i izlaznog sloja. Veći broj slojeva omogućava dubokim neuronskim mrežama učenje kompleksnijih mapiranja  $y = f(x; \theta)$ . Nadalje, svi algoritmi i pravila opisana u prethodnom odlomku vrijede i za duboke neuronske mreže. Moderne primjene dubokih neuronskih mreža uključuju obradu prirodnog jezika, strojno prevođenje, bioinformatiku, računalni vid i mnoge druge. Za ovaj rad ključna je njihova primjena u području računalnog vida. Vrsta dubokih neuronskih mreža koja je izuzetno bitna za to područje su konvolucijske neuronske mreže te su iste pobliže opisane u sljedećem odlomku.

## 2.3. Konvolucijske neuronske mreže

### 2.3.1. Uvod

Konvolucijske neuronske mreže (engl. *Convolutional neural network*) temelj su modernog računalnog vida. Radi se o neuronskim mrežama koje, u barem jednom, sloju umjesto tradicionalnog matričnog množenja primjenjuju operaciju konvolucije [6]. U prvom dijelu odlomka opisat ćemo operaciju konvolucije, u drugom dijelu ćemo detaljnije opisati konvolucijske slojeve te ćemo se za kraj dotaknuti još jednog važnog sloja u konvolucijskim mrežama, sloja sažimanja.

### 2.3.2. Matematička konvolucija

Konvolucija je matematička operacija nad dvije funkcije koja rezultira trećom funkcijom. Treća funkcija opisuje količinu preklapanja između dvije funkcije dok se jedna od njih pomiče po domeni. Takva definicija vidljiva je i iz jednadžbe 2.12 koja je standardna formula konvolucije za kontinuirane funkcije  $f$  i  $g$ . Nadalje, konvoluciju predstavljamo znakom  $*$ .

$$z(t) = (f * g)(t) = \int f(\tau)g(t - \tau)d\tau \quad (2.12)$$

Kako u radu s računalima susrećemo većinom diskretne podatke u jednadžbi 2.13 opisana je definicija operacije konvolucije za diskretne funkcije  $f$  i  $g$ .

$$z(t) = (f * g)(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(t - \tau) \quad (2.13)$$

U terminologiji konvolucijskih mreža funkcija  $f$  bi predstavljala ulazne podatke, funkcija  $g$  jezgru, a izlaz operacije naziva se mapa značajki [6]. Ovako opisanu konvoluciju možemo primijeniti i na dvodimenzionalne podatke kao što je slika  $I$ .

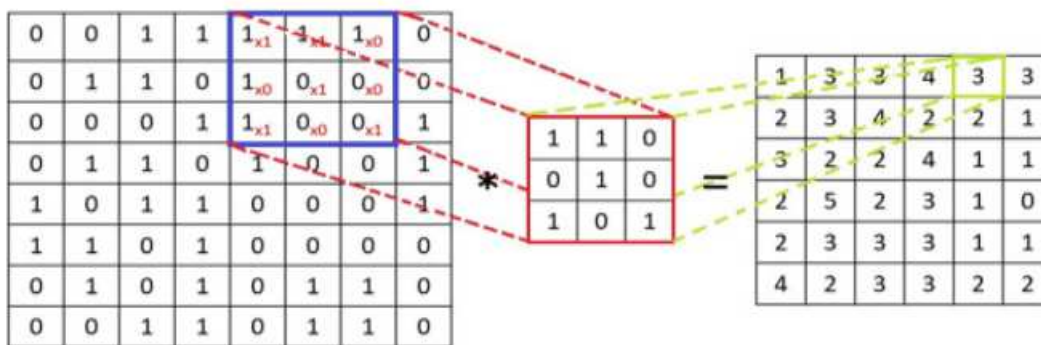
$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(m - i, n - j) \quad (2.14)$$

U praksi, većina biblioteka za duboko učenje implementira povezanu funkciju korelacije 2.15 koja je identična konvoluciji izuzev zrcaljena jezgre [6].

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m + i, n + j)K(m, n) \quad (2.15)$$

### 2.3.3. Konvolucijski sloj

Temelj konvolucijskih slojeva, i ono što ih čini idealnima za rad sa slikama, njihova je rijetka povezanost s ulaznim podacima. Aktivacije konvolucijskog sloja povezane su samo s odgovarajućim lokalnim susjedstvom ulaza u sloj. Aktivacije konvolucijskih slojeva organizirane su u mape značajki i sve aktivacije u istoj mapi značajki dijele parametre. Trenutno razmatramo konvolucijski sloj s jednom mapom značajki na ulazu i izlazu ali kasnije ćemo to poopćiti na slučaj s više mapa značajki. Posljednji i najbitniji element konvolucijskog sloja je jezgra  $K$ . Veličina jezgre određuje veličinu lokalnog susjedstva s kojom je povezan neuron u trenutnom konvolucijskom sloju kao što je vidljivo iz slike 2.3. Također, jezgra je parametar koji učimo u konvolucijskim slojevima te je zapravo ekvivalent matricama težina u potpuno povezanim slojevima.



**Slika 2.3:** Prikaz konvolucijskog sloja s jezgrom veličine  $3 \times 3$ . Slika je preuzeta iz [16].

U praksi, mape značajki su višedimenzionalne te u jednom konvolucijskom sloju imamo više jezgri od kojih svaka odgovara jednom sloju mape značajki. Dimenzije jednog sloja mape značajki jednoznačno su određene s veličinom ulaza, veličinom jezgre i veličinom koraka pomaka jezgre. Za kraj bitno je primijetiti kako ime mapa značajki potječe upravo iz onoga što konvolucijski slojevi rade. Naime, konvolucijski slojevi detektiraju značajke na slikama, kao što su na primjer rubovi. Slijedno tome više uzastopnih konvolucijskih slojeva može otkriti sve apstraktnije i apstraktnije značajke. Većina modernih arhitektura konvolucijskih neuronskih mreža služi se upravo prije opisanim pristupom, slika na ulazu prolazi kroz više uzastopnih konvolucijskih slojeva uz poneki sloj sažimanja te na kraju izlazi iz posljednjeg konvolucijskog sloja prolaze kroz jedan ili više potpuno povezanih slojeva za klasifikaciju.

### 2.3.4. Sloj sažimanja

Sloj sažimanja jedan je od ključnih elemenata svih modernih arhitektura konvolucijskih neuronskih mreža. Primjenjujemo ga nakon konvolucijskog sloja kako bi smanjili veličinu mape značajki. Jednostavno rečeno, sloj sažimanja zamjenjuje neki izlaz konvolucijskog sloja sa statistikom okolnih izlaza [6]. Za veličinu lokalnih izlaza koje uzimamo najčešće se postavlja kvadrat veličine  $2 \times 2$  kojeg pomičemo po izlazima koracima veličine jedan ili dva. Nadalje, naziv sloja sažimanja kojeg koristimo jednoznačno je određen statistikom koja zamjenjuje izlaz. Najčešće korišteni slojevi sažimanja su maksimalni sloj sažimanja i prosječni sloj sažimanja. Svrha sloja sažimanja je smanjenje utjecaja malih translacija u ulaznim podacima na reprezentaciju značajki ulaza u mapama značajki kakve gradimo konvolucijskim slojevima.



## 3. Inkrementalno učenje uz dodavanje novih razreda

### 3.1. Uvod

Inkrementalno učenje uz dodavanje novih razreda predstavlja jedan od glavnih izazova dubokog učenja u sustavima za rad u stvarnom vremenu [18]. Trenutna praksa u dubokom učenju za raspoznavanje slika treniranje je modela na cijelom skupu podataka. Drugim riječima, svi razredi koje model treba naučiti raspoznati poznati su prilikom treniranja. Algoritmi inkrementalnog učenja uz dodavanje novih razreda razmatraju realan scenarij u kojem nam nisu unaprijed poznati svi razredi koje će model morati naučiti raspoznati. U svim algoritmima inkrementalnog učenja pretpostavljamo postojanje modela naučenog na starim podacima. Cilj inkrementalnog učenja dobivanje je modela koji je primjenjiv i na nove i na stare podatke te raspoznaje i stare i nove razrede. Glavni izazov algoritmima inkrementalnog učenja predstavlja problem katastrofalnog zaboravljanja (engl. *catastrophic forgetting*) [5] [13]. Katastrofalno zaboravljanje definiramo kao veliki gubitak točnosti neuronske mreže na starom zadatku nakon što je istrenirana na novom zadatku. Objavljivanje jedne od pionirskih metoda inkrementalnog učenja  $LwF$  [11] i danas najpoznatije metode inkrementalnog učenja iCARL [14] dovelo je do razvoja brojnih drugih algoritama kroz narednih nekoliko godina. Općenita podjela pristupa inkrementalnom učenju može se napraviti ovisno o memorijskom budžetu metode. Memorijski budžet količina je podataka, u našem slučaju slika, koje spremamo iz originalnog skupa podataka za dodatno poboljšavanje modela. Prilikom svakog sljedećeg koraka inkrementalnog učenja biramo slike novih razreda koje ćemo dodati u memorijski budžet te izbacujemo odgovarajući broj slika starih razreda. Broj slika svakog razreda u memorijskom budžetu ovisi o njegovoj veličini te broju razreda koje model treba raspoznati ali on mora biti jednak za sve razrede. S druge strane, većina današnjih algoritama može raditi sa i bez memorijskog budžeta. Za bolju podjelu okrenuo sam se radu [2] koji dijeli algoritme inkremental-

nog učenja u tri skupine koje ću pobliže opisati u narednom odlomku. Te skupine čine algoritmi bazirani na rastu modela, algoritmi bazirani na fiksiranom prikazu i algoritmi bazirani na finom podešavanju. Nadalje, u radu [2] razrađeno je 6 poželjnih svojstava algoritama inkrementalnog učenja te ću prilikom opisivanja skupina algoritama veliku pažnju posvetiti upravo tim svojstvima. Svojstva su sljedeća:

- **Kompleksnost** - mogućnost integracije novih klasa bez povećanja složenosti konačnog modela.
- **Memorija** - zahtjeva li algoritam postojanje memorijskog budžeta za postizanje zadovoljavajućih performansi.
- **Točnost** - ima li konačni model zadovoljavajuću točnost.
- **Vrijeme** - koliko vremena je potrebno za integraciju novo dodanih razreda u konačni model.
- **Plastičnost** - prilagođava li se model dobro novim podacima.
- **Skalabilnost** - mogućnost kvalitetnog učenja kroz puno koraka inkrementalnog učenja i primjenjivost u stvarnom svijetu.

## 3.2. Podjela pristupa

### 3.2.1. Algoritmi bazirani na rastu modela

Algoritmi bazirani na rastu modela (engl. *Model-growth based*) povećavaju veličinu dubokih modela kako bi uključili novo znanje. Modeli ovog pristupa jako se dobro prilagođavaju novim podacima i ne zahtijevaju memorijski budžet za spremanje starih podataka, ako sustav ima dovoljno memorije za podržavanje rasta modela. Njihova konačna točnost ovisi o dopuštenom povećanju modela. S druge strane, modeli izgrađeni ovim algoritmima imaju visoku kompleksnost, zbog nužnog rasta modela, kako bi naučili predviđati nove razrede. Nadalje, sa svakim uvođenjem novih razreda potrebno je ponovno treniranje modela. Za ovu skupinu algoritama ključno je naći ravnotežu između konačne točnosti i kompleksnosti kranjeg modela [2].

### 3.2.2. Algoritmi bazirani na fiksiranom prikazu

Algoritmi bazirani na fiksiranom prikazu (engl. *Fixed-representation based*) fiksiraju težine modela nakon prvog neinkrementalnog koraka te kasnije ponovno treniraju samo posljednji klasifikacijski sloj. Ovaj pristup inkrementalnom učenju manje

je zastupljen u literaturi zbog slabijih performansi u odnosu na ostala dva pristupa. Nadalje, performanse jako ovise o kvaliteti početne reprezentacije cijelog skupa podataka, što je za problem inkrementalnog učenja prilično jak uvjet. S druge strane, ovi algoritmi imaju konstantnu kompleksnost modela i ne zahtijevaju ponovno treniranje cijelog modela na novim podacima. Algoritmi bazirani na fiksiranom prikazu jedini su pristup koji ne zahtijeva ponovno treniranje te to omogućuje brzu integraciju novog znanja. Isto tako, zbog fiksiranih težina modela, algoritmi ovog pristupa ne iziskuju memorijski budžet za spremanje slika originalnog skupa podataka [2].

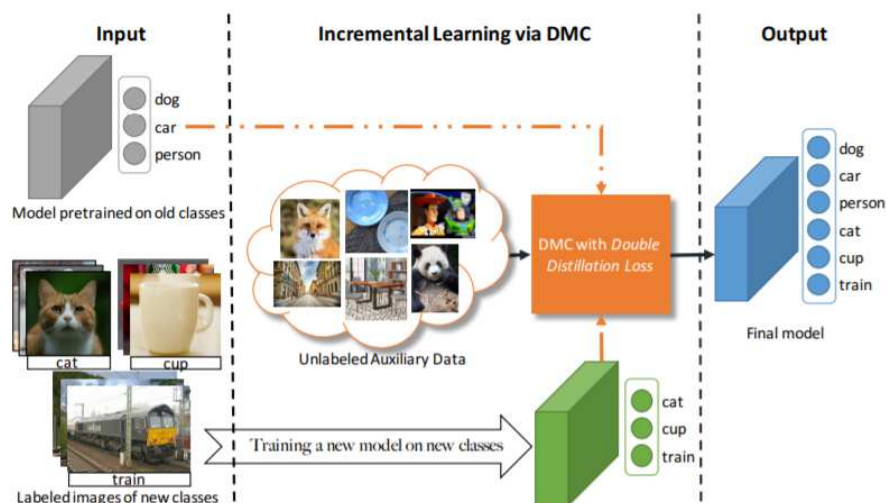
### **3.2.3. Algoritmi bazirani na finom podešavanju**

Većina dostupnih algoritama za inkrementalno učenje može se svrstati pod algoritme bazirane na finom podešavanju (engl. *Fine-tuning based*). Najbolje rezultate u ovom pristupu postižu algoritmi bazirani na destilaciji znanja (engl. *knowledge distillation*), pretvaranju inkrementalnog učenja u problem neuravnoteženog učenja (engl. *imbalanced learning*) što je problem postojanja značajne razlike u broju slika svakog razreda, ili kombinaciji tog dvoje. Zastupljenost ovih algoritama može se objasniti njihovom generalno jako dobrom točnošću za inkrementalno učenje i konstantnoj kompleksnosti krajnjeg modela. Ove dobre karakteristike dolaze nauštrb brzini dobivanja novog modela jer svako uvođenje novih razreda zahtijeva ponovno treniranje modela. Nadalje, većina algoritama ovog pristupa jako je ovisna o memorijskom budžetu i pokazuje značajan pad u performansama ako broj razreda postane prevelik za kvalitetnu reprezentaciju svakog razreda u dostupnom memorijskom budžetu. Ako sustav to dozvoljava, idealno bi bilo postojanje rastućeg memorijskog budžeta koji zadržava konstantan broj slika iz pojedinog razreda u memoriji [2]. Postoje metode koje pokušavaju riješiti prije navedene probleme vezane uz memorijski budžet te je jedna od njih konsolidacija dubokih modela koja je glavna tema ovog rada i detaljnije je opisana u narednom poglavlju.

## 4. Konsolidacija dubokih modela

### 4.1. Uvod

Konsolidacija dubokih modela relativno je nova metoda inkrementalnog učenja uz dodavanje novih razreda. Prvi put opisana je u radu [20] te spada pod algoritme bazirane na finom podešavanju. Metoda se općenito sastoji od dvije faze uz treću fazu koja nije nužna ali povećava krajnju točnost. U prvoj fazi treniramo novi model na novim razredima. U sljedećoj fazi, koja je temelj ove metode, konsolidiramo stari i novi model koristeći neoznačene pomoćne slike kao što je opisano na slici 4.1. Pojam konsolidacija odnosi se na prijenos znanja s dva modela, od kojih jedan raspoznaje stare, a drugi nove razrede, na novi treći model koji će raspoznavati i stare i nove razrede. U trećoj fazi koristimo memorijski budžet sa slikama iz originalnog skupa podataka kako bi dodatno trenirali model dobiven konsolidacijom. U originalnom radu [20] konsolidacija dubokih modela predložena je kao metoda koja ne zahtijeva memorijski budžet. Tako opisana metoda izbjegava probleme s privatnošću i s nemogućnošću pohrane starih podataka. S druge strane, korištenje podskupa starih podataka za dodatno treniranje obično donosi bolje performanse te je zato predloženo kao moguća nadogradnja konsolidacije dubokih modela. Upravo je istraživanje utjecaja dodavanja dodatnog treniranja s primjerima slika iz originalnog skupa podataka glavni doprinos ovog rada. U nastavku ovog poglavlja detaljnije je opisana svaka od tri prethodno spomenute faze.



**Slika 4.1:** Opis rada konsolidacije dubokih modela. Nakon treniranja modela za raspoznavanje novih razreda konsolidiramo taj model s modelom koji zna raspoznavati stare razrede. Konsolidacija podrazumijeva prijenos znanja s dva prethodno opisana modela na novi treći model koji će znati raspoznavati i stare i nove razrede. Prijenos znanja ostvarujemo korištenje dvostrukog destilacijskog gubitka. Dvostruki destilacijski gubitak je funkcija gubitka korištena prilikom konsolidacije. Temelji se na kvadratnom gubitku, ali umjesto očekivanog izlaza imamo spojene izlaze dva prethodno spomenuta modela. Slika je preuzeta iz [20]

## 4.2. Faza treniranja novog modela

Prilikom treniranja novog modela provodimo standardno treniranje s kraja na kraj na svim dostupnim slikama za nove razrede koje uvodimo. Dobivamo model  $f_{novi}$  koji je naučen prepoznavati nove razrede. U sljedećoj fazi cilj nam je iz  $f_{novi}$  i  $f_{stari}$  dobiti finalni model  $f$ . Za ovu metodu sama arhitektura modela  $f_{novi}$  je nebitna te je u idućoj fazi moguće konsolidirati  $f_{novi}$  s  $f_{stari}$  bez obzira na njihovu arhitekturu.

## 4.3. Faza konsolidacije

Po ovoj fazi je konsolidacija dubokih modela dobila svoje ime. U njoj se koristi varijacija destilacije znanja [8] što je popularna tehnika za prijenos znanja s jedne mreže na drugu. Originalno, destilacija znanja predložena je za prijenos znanja s kompleksne na manje kompleksnu mrežu. Za potrebe inkrementalnog učenja u [20] opisan je dvostruki destilacijski gubitak (engl. *double distillation loss*) koji omogućava des-

tilaciju znanja s dvije mreže na jednu. Prilikom konsolidacije ne koristimo slike iz originalnog skupa podataka već neoznačene slike koje imaju sličnu kompleksnost kao i slike koje model mora naučiti raspoznati. S tim slikama prvo napravimo unaprijedni prolaz kroz  $f_{novi}$  i  $f_{stari}$  te uzmemo izlaze iz posljednjeg sloja prije nego što je propušten kroz softmax aktivacijsku funkciju. Time dobivamo izlaze  $\hat{y}_{stari} = [\hat{y}^1, \dots, \hat{y}^k]$  i  $\hat{y}_{novi} = [\hat{y}^{k+1}, \dots, \hat{y}^n]$  gdje eksponenti predstavljaju klase. Slijedeći praksu iz [20] koristio sam kvadratni gubitak kao funkciju duplog destilacijskog gubitka. S tim je optimizacijski cilj definiran kao u jednadžbi 4.1 gdje je  $F$  faktor dijeljenja gubitka koji služi za sprječavanje eksplodirajućih gradijenata,  $y_i$  predikcija konsolidiranog modela, a  $\hat{y}_i$  spoj izlaza iz  $f_{novi}$  i  $f_{stari}$  koji se koristi za učenje novog modela.

$$\min_{\theta} \frac{1}{F} \sum_{x_i \in U} L_{dd}(y_i, \hat{y}_i) \quad (4.1)$$

Dupli destilacijski gubitak definiramo uz pomoć kvadratnog gubitka.

$$L_{dd}(y, \hat{y}) = \frac{1}{t} \sum_{j=1}^t (y^j - \hat{y}^j)^2 \quad (4.2)$$

U jednadžbi 4.2  $y^j$  predstavlja izlaz iz konsolidiranog modela za razred  $j$ , a  $\hat{y}^j$  spoj je izlaza iz  $\hat{y}_{novi}$  i  $\hat{y}_{stari}$  po jednadžbi 4.3.

$$\hat{Y}^j = \begin{cases} \hat{y}^j - \frac{1}{s} \sum_{k=1}^s y^k & 1 \geq j \leq s \\ \hat{y}^j - \frac{1}{t-s} \sum_{k=s}^t y^k & s < j \leq t \end{cases} \quad (4.3)$$

U jednadžbi 4.3 normaliziramo izlaze iz pripadajućih modela oduzimajući od njih srednju vrijednost svih izlaza tog modela. Taj korak služi kako bi doveli logite dva modela na istu skalu te kako konsolidirani model ne bi pokazivao pristranost prema jednom od dva modela [20].

Nadalje, bitan je izbor neoznačenih pomoćnih slika te je u [20] pokazano kako konsolidacija neće raditi za bilo koji izbor. Ja sam se odlučio za korištenje ImageNet32 skupa podataka jer konsolidacija njime pokazuje stabilne i dobre performanse. ImageNet32 detaljnije je opisan u sljedećem poglavlju.

Bitno je naglasiti kako novi model koji konsolidiramo ne smijemo inicijalizirati ni iz jednog od dva modela koja imamo već nasumično.

## 4.4. Faza finog podešavanja

Fino podešavanje (engl. *Fine-tuning*) proizvoljna je treća faza konsolidacije dubokih modela. U slučaju kada implementiramo ovu fazu koristimo memorijski budžet te dodatno treniramo konsolidirani model na primjercima originalnog skupa slika. U izvornom radu [20] ova faza nije implementirana kao dio konsolidacije dubokih modela već je spomenuta u dodatku D. kao moguće proširenje. Glavni cilj mog rada je kvantifikacija utjecaja ove faze na konačnu točnost modela. Nadalje, bitan fokus stavio sam na utjecaj ove faze na različite eksperimentalne postavke.

# 5. Implementacija

## 5.1. Okvir za automatsko diferenciranje

PyTorch je biblioteka otvorenog koda zasnovana na biblioteci Torch. Razvijena je primarno od strane FAIR-a (Facebook's AI Research lab) te se često koristi u području računalnog vida. Primarna struktura podataka u biblioteci PyTorch su tenzori. PyTorch omogućuje značajno ubrzanje rada s tenzorima kroz integraciju s grafičkim procesorom te je modeliranje neuronskih mreža jednostavno i intuitivno. Iz svih prije navedenih razloga odabrao sam PyTorch kao alat za automatsku diferencijaciju u svim provedenim eksperimentima.

## 5.2. Skupovi podataka

### 5.2.1. CIFAR-100

CIFAR-100 skup je slika koji se sastoji od 60000 32x32 slika u boji koje su ravnomjerno raspodijeljene u 100 razreda. Skup slika dodatno je podijeljen u skup za treniranje, koji se sastoji od 50000 slika, i skup za testiranje s preostalim 10000. Skup za treniranje i skup za testiranje sadrže jednak broj slika svakog razreda [9]. Korištenje CIFAR-100 skupa slika standard je za evaluaciju algoritama inkrementalnog učenja uz dodavanje novih razreda već nekoliko godina od kada je prvi put predložen u [14].

### 5.2.2. ImageNet32

ImageNet32 skup slika verzija je originalnog ImageNet skupa slika u kojem su sve slike smanjene na veličinu 32x32. Samim time, ImageNet32 sadrži jednak broj slika kao i ImageNet, 1281167 slika u skupu za treniranje i 50000 slika u skupu za testiranje. [4]. ImageNet32 predložen je u [20] kao idealan izvor slika za konsolidaciju modela za raspoznavanje slika iz prije opisanog skupa CIFAR-100. Za razliku od njihovog pris-



tupa ova implementacija koristi samo skup za testiranje što značajno smanjuje potrebni memorijski budžet što je u [12] istaknuto kao jedan od glavnih problema s konsolidacijom dubokih modela.

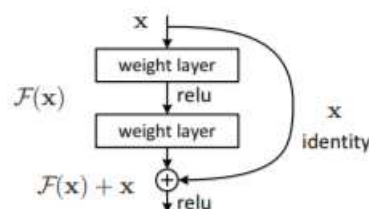
### 5.2.3. Uvećanje podataka

Za uvećanje podataka (engl. *data augmentation*) vodio sam se praksom opisanom u [7] i [20]. Na svaku sliku iz skupa za treniranje nadodao sam 4 piksela nadopune sa svake strane. Zatim sam nasumično uzeo okvir veličine  $32 \times 32$  iz dobivene slike dimenzija  $36 \times 36$  ili njene vodoravne preslike.

## 5.3. Arhitektura modela

### 5.3.1. Rezidualne mreže

Rezidualne mreže vrsta su konvolucijskih neuronskih mreža koja je prvi put opisana u [7] te je prepoznatljiva po rezidualnim blokovima s preskočnim vezama (engl. *shortcut connections*). Jedan rezidualni blok prikazan je na slici 5.1 te se ResNet-32, koji je korišten u svim eksperimentima, sastoji od 15 takvih blokova. Nadalje, rezidualne mreže sadrže jedan ulazni konvolucijski sloj i jedan izlazni potpuno povezani sloj za klasifikaciju.



Slika 5.1: Rezidualni blok [7].

Uvođenje rezidualnih mreža omogućilo je izgradnju dubljih modela nego što je prije bilo korisno. Naime, kao što je opisano u [7], uvođenje dodatnih slojeva u običnu konvolucijsku mrežu ne rezultira boljom točnošću nakon određene dubine. S druge strane, rezidualne mreže rješavaju taj problem i ostvarile su zamjetne rezultate na mnogim standardnim skupovima podataka te su postale standard za primjene u računalnom vidu.

## 5.4. Hiperparametri i detalji treniranja

Kod postavljanja hiperparametara vodio sam se postavkama predloženima u [20] koliko je to bilo moguće. Neke hiperparametre bilo je potrebno optimizirati za specifičnu konfiguraciju te su oni navedeni u pripadajućim odlomcima konfiguracija u narednom poglavlju. U svim eksperimentima i fazama treniranja koristio sam SGD optimizator dostupan u biblioteci PyTorch s momentom postavljenim na 0.9 i prigušenjem težina (engl. *weight decay*) postavljenim na 0.0001. Prije treniranja redosljed razreda izmiješan je kako bi se smanjio utjecaj redoslijeda na postupak inkrementalnog učenja. Nadalje, u fazi treniranja novih modela i fazi konsolidacije množio sam korak učenja s 0.1 na 70% i 90% svih epoha te je veličina mini grupe postavljena na 128. U fazi treniranja novih modela broj epoha postavio sam na 100, a u fazi konsolidacije na 50. Za konsolidaciju modela koristio sam 16000 nasumično izabranih slika iz prije opisanog ImageNet32 skupa za testiranje.

Nakon svake konsolidacije na dobivenom modelu bih proveo fino podešavanje sa spremjenim podskupom originalnog skupa podataka. Memorijski budžet sam postavio na  $K = 2000$  slijedeći praksu opisanu u [14]. U tih 2000 slika nalazi se jednak broj slika od svakog dosad naučenog razreda. Nakon nasumičnog izbora tih 2000 slika provodi se fino podešavanje kroz 15 epoha s korakom učenja 0.001. Veličina mini grupe postavljena je na cijeli skup podataka.

## 6. Eksperimenti

U nastavku prikazani su rezultati eksperimenata inkrementalnog učenja uz dodavanjem novih razreda primjenom konsolidacije dubokih modela. Zajednički detalji provedenih eksperimenata opisani su u prethodnim poglavljima, a detalji ovisni o konfiguraciji bit će opisani na početku svakog odlomka. Eksperimenti su podijeljeni u četiri općenite konfiguracije koje se razlikuju ovisno o broju klasa na kojima model uči u jednoj iteraciji inkrementalnog učenja. Preciznije, skup podataka CIFAR-100 podijeljen je na podskupove od 50, 20, 10 i 5 klasa ovisno o konfiguraciji te je takva podjela u skladu s eksperimentalnim postavkama opisanima u [14], [20] i [2]. Nadalje, korištena je standardna notacija za konfiguracije u obliku  $M/N$  gdje  $M$  predstavlja broj naučenih razreda u početnom modelu, a  $N$  broj novododanih razreda u svakom koraku inkrementalnog učenja. U tablici 6.1 prikazane su krajnje točnosti modela nakon konsolidacije i finog podešavanja sa sačuvanim primjercima slika iz originalnog skupa podataka. Rezultati će biti detaljnije analizirani za svaku konfiguraciju u pripadajućem odlomku. Kao dodatak analizi u svakom odlomku biti će prikazani i rezultati treniranja samo na memorijskom budžetu kako bi se pokazao doprinos inkrementalnog učenja u odnosu na nadzirano učenje.

**Tablica 6.1:** Rezultati krajnje točnosti za sve konfiguracije. Korištena je standardna notacija za konfiguracije u obliku  $M/N$  gdje  $M$  predstavlja broj naučenih razreda u početnom modelu, a  $N$  broj novododanih razreda u svakom koraku inkrementalnog učenja. Prikazani su rezultati nakon konsolidacije i nakon finog podešavanja.

Konfiguracija	Točnost finalnog modela nakon konsolidacije [%]	Točnost finalnog modela nakon konsolidacije i finog podešavanja[%]
100	-	65.22
50/50	25.22	32.23
20/20	12.29	20.17
10/10	18.67	20.34
5/5	16.23	18.77

Slijedeći praksu iznesenu u [14] u tablici 6.2 iznio sam i prosječne točnosti kroz inkrementalne korake. Krajnja točnost koja je iznesena u tablici 6.1 jako ovisi o konfiguraciji jer se točnost nužno smanjuje s velikim brojem inkrementalnih koraka kao što je slučaj u konfiguraciji 5/5. Iz tog razloga, prosječna točnost kroz inkrementalne korake bolja je za usporedbu različitih konfiguracija. Iz tablice 6.2 vidimo kako je raspon prosječnih točnosti 4.84 postotnih bodova. Takav raspon indikacija je kako konsolidacija dubokih modela nije previše ovisna o odabranoj konfiguraciji.

**Tablica 6.2:** Rezultati prosječne točnosti kroz inkrementalne korake za sve konfiguracije. Korištena je standardna notacija za konfiguracije u obliku  $M/N$  gdje  $M$  predstavlja broj naučenih razreda u početnom modelu, a  $N$  broj novododanih razreda u svakom koraku inkrementalnog učenja. Prikazani su rezultati nakon konsolidacije i nakon finog podešavanja.

Konfiguracija	Prosječna točnost modela nakon konsolidacije [%]	Prosječna točnost modela nakon konsolidacije i finog podešavanja [%]
100	-	65.22
50/50	25.22	32.23
20/20	23.33	30.60
10/10	30.55	34.11
5/5	32.21	35.44

## 6.1. Konfiguracija 50/50

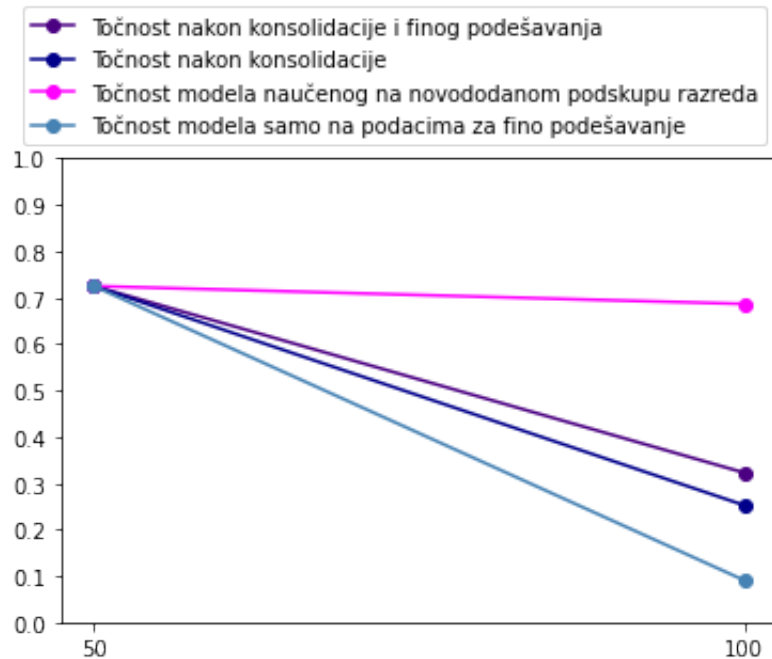
Ova konfiguracija predstavlja najmanji izazov s obzirom na to kako konačni model konsolidiramo samo jednom. Takvo razmišljanje opravdavaju i rezultati koji su prikazani u tablici 6.1. Iz slike 6.1 vidimo kako permutirani poredak klasa, zbog većih podskupova, ovoj konfiguraciji ne predstavlja toliko problem prilikom treniranja modela za prepoznavanje novih klasa. Eksperimentalne postavke specifične za ovu konfiguraciju nalaze se u tablici 6.3.

**Tablica 6.3:** Eksperimentalne postavke za konfiguraciju 50/50.

Naziv parametra	Vrijednost parametra
Korak učenja u fazi treniranja novog modela	0.1
Korak učenja u fazi konsolidacije	0.1
Faktor dijeljenja gubitka	100

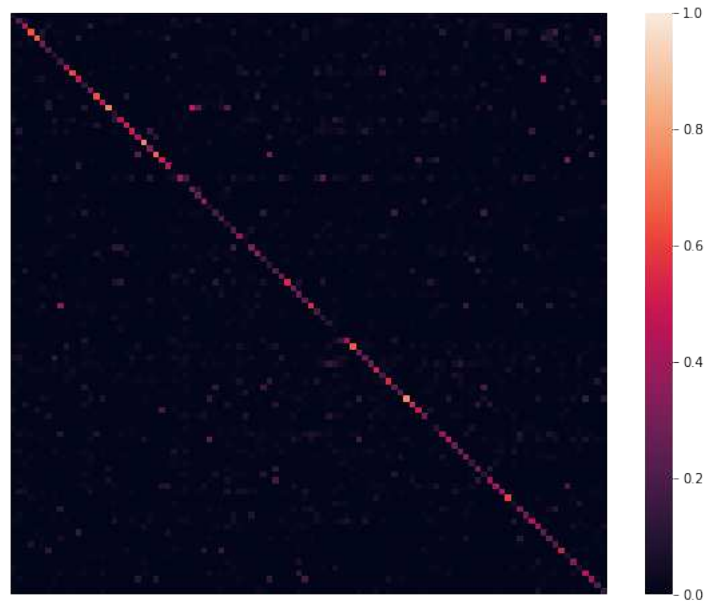
Nadalje, pokazuje se kako metoda konsolidacije dubokih modela pruža bolje re-

zultate u ovoj konfiguraciji nakon faze finog podešavanja. Iz tablice 6.1 vidimo kako je točnost nakon konsolidacije 25.22%, a nakon konsolidacije i finog podešavanja 32.23%. Razlika u točnosti je 7.01% što nije zanemarivo te indicira na korisnost spremanja određene količine slika iz originalnog skupa podataka, naravno, ako pravila o privatnosti to dozvoljavaju. Slika 6.1 nam također pokazuje doprinos inkrementalnog učenja u odnosu na učenje samo na memorijskom budžetu za fino podešavanje.



**Slika 6.1:** Točnosti inkrementalnog treniranja modela na konfiguraciji 50/50. Na osi apscisa prikazani su koraci inkrementalnog učenja.

Za kraj, potrebno je uzeti u obzir matricu zabune čime provjeravamo ako naš model ima problem katastrofalnog zaboravljanja. Na slici 6.2 vidimo kako konsolidacija dubokih modela performira jako dobro te ne pokazuje intrinzičnu pristranost prema starim ili novim klasama, unatoč tome što zbog relativno male točnosti ima puno krivih predviđanja.



**Slika 6.2:** Matrica zabune za konfiguraciju 50/50. Svaki redak i pripadajući stupac označavaju jedan razred. Redci predstavljaju predviđanja modela, a stupci točne razrede.

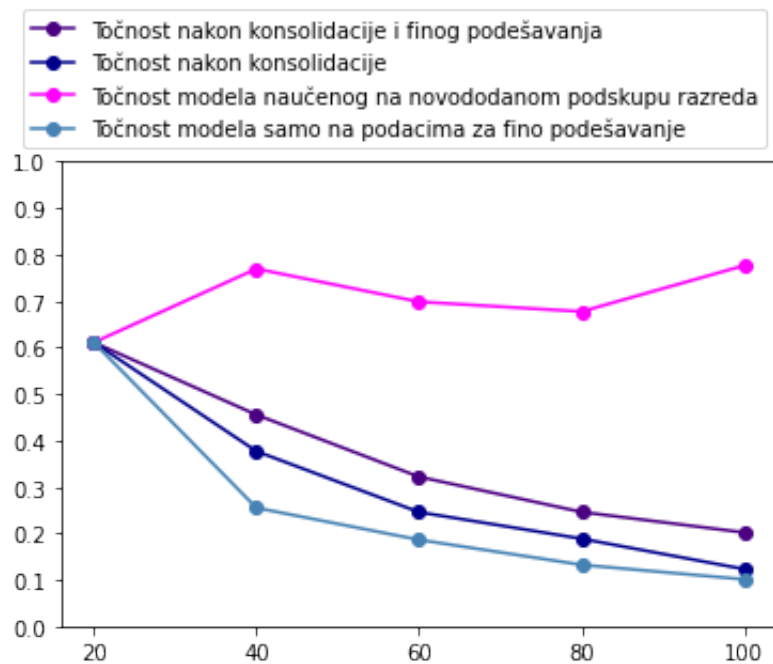
## 6.2. Konfiguracija 20/20

U ovoj konfiguraciji možemo nazrijeti problem inherentne težine nekih kombinacija klasa u podskupovima. Raspon točnosti modela treniranih na novim klasama je čak 16.55 postotnih bodova. Unatoč tome, metoda konsolidacije pokazuje stabilnu izvedbu kao što vidimo iz slike 6.3.

**Tablica 6.4:** Eksperimentalne postavke za konfiguraciju 20/20.

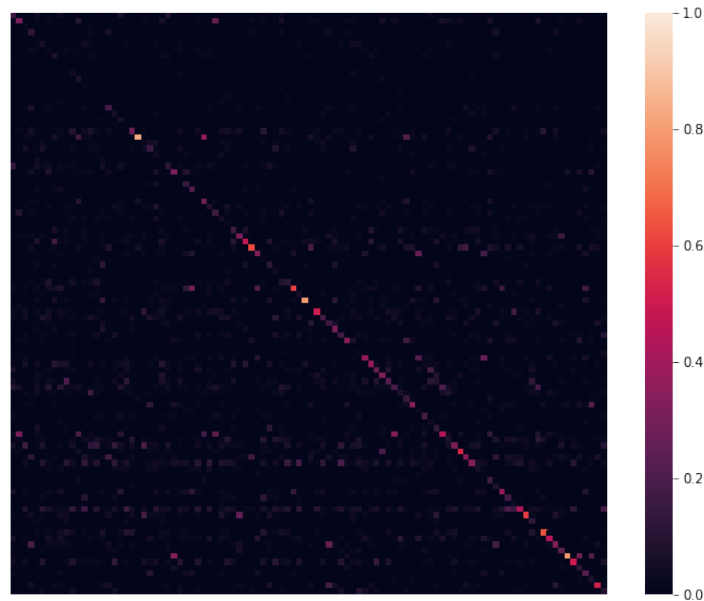
Naziv parametra	Vrijednost parametra
Korak učenja u fazi treniranja novog modela	0.1
Korak učenja u fazi konsolidacije	0.01
Faktor dijeljenja gubitka	20

Prosječni dobitak u točnosti nakon faze finog podešavanja je 7.27%, što pokazuje kako je fazu finog podešavanja poželjno koristiti i u ovoj konfiguraciji. S druge strane, doprinos inkrementalnog učenja za ovu konfiguraciju manji je nego za konfiguraciju 50/50. To je i za očekivati jer ova konfiguracija ima nižu krajnju točnost, a krajnja točnost učenja samo na memorijskom budžetu je relativno konstantna.



**Slika 6.3:** Točnosti inkrementalnog treniranja modela na konfiguraciji 20/20. Na osi apscisa prikazani su koraci inkrementalnog učenja.

Iz matrice zabune na slici 6.4 vidimo kako je prisutno više ulaza koji su veći od nule u odnosu na sliku 6.2. Takav rezultat je očekivan te je posljedica veće zahtjevnosti ove konfiguracije. Ključno je kako iz slike 6.4 i dalje možemo vidjeti kako ne postoji pristranost prema klasama što se pokazuje kao jedna značajna snaga ove metode.



**Slika 6.4:** Matrica zabune za konfiguraciju 20/20. Svaki redak i pripadajući stupac označavaju jedan razred. Redci predstavljaju predviđanja modela, a stupci točne razrede.

### 6.3. Konfiguracija 10/10

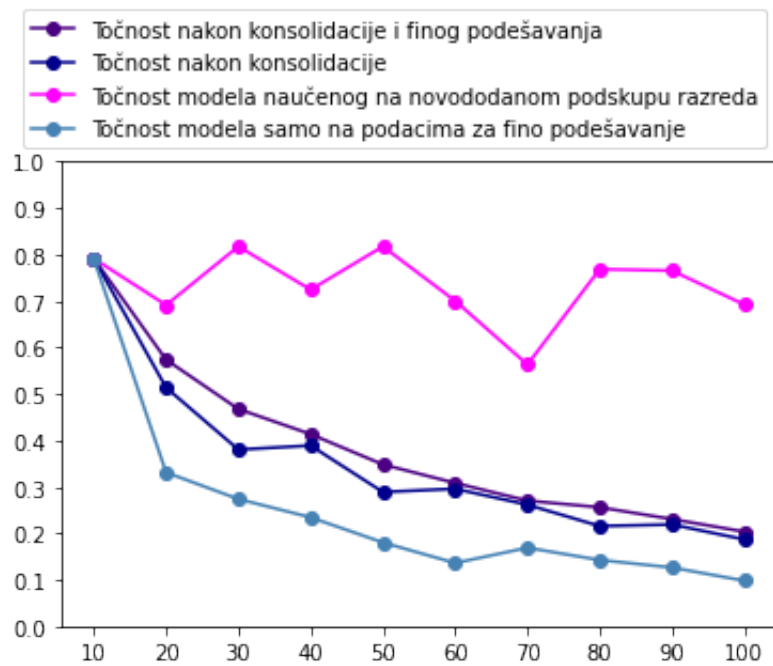
U treniranju novih modela u konfiguraciji 10/10 raspon točnosti još jasnije pokazuje inherentnu težinu kombinacije nekih klasa. Taj raspon u ovoj konfiguraciji iznosi 25.4%. Smanjenje tog raspona bilo bi poželjno, ali konačna točnost modela ostaje stabilna unatoč oscilacijama točnosti novih modela što je vidljivo iz slike 6.5.

**Tablica 6.5:** Eksperimentalne postavke za konfiguraciju 10/10.

Naziv parametra	Vrijednost parametra
Korak učenja u fazi treniranja novog modela	0.01
Korak učenja u fazi konsolidacije	0.01
Faktor dijeljenja gubitka	10

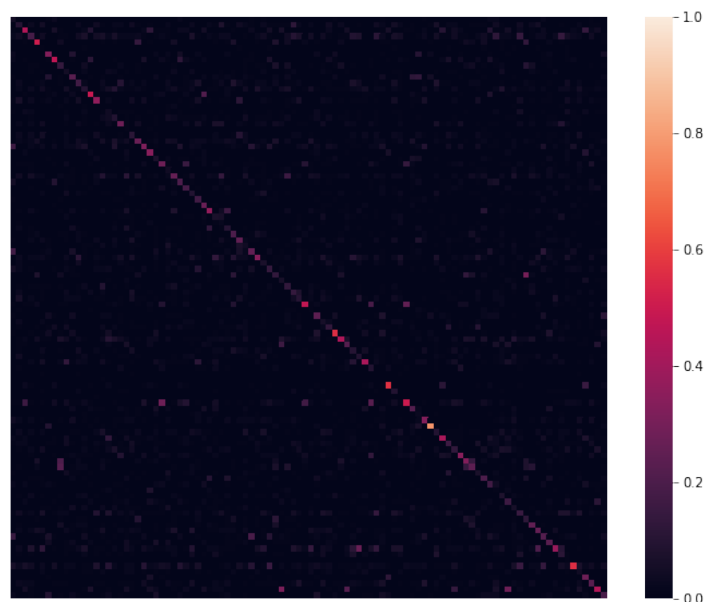
S druge strane, u ovoj konfiguraciji primjećujemo pad povećanja točnosti između faza konsolidacije i finog podešavanja. Pad povećanja točnosti vidljiv je i na slici 6.5, a prosječna vrijednost povećanja točnosti iznosi 3.57%. Ta vrijednost značajno je manja nego u prethodne dvije konfiguracije i može biti indikacija kako spremanje primjeraka originalnog skupa podataka u ovoj konfiguraciji nije toliko korisno. Doprinos inkrementalnog učenja i dalje postoji te opravdava korištenje istoga.





**Slika 6.5:** Točnosti inkrementalnog treniranja modela na konfiguraciji 10/10. Na osi apscisa prikazani su koraci inkrementalnog učenja.

Matrica zabune na slici 6.6 za ovu konfiguraciju nastavlja trend nepostojanja pristranosti modela.



**Slika 6.6:** Matrica zabune za konfiguraciju 10/10. Svaki redak i pripadajući stupac označavaju jedan razred. Redci predstavljaju predviđanja modela, a stupci točne razrede.

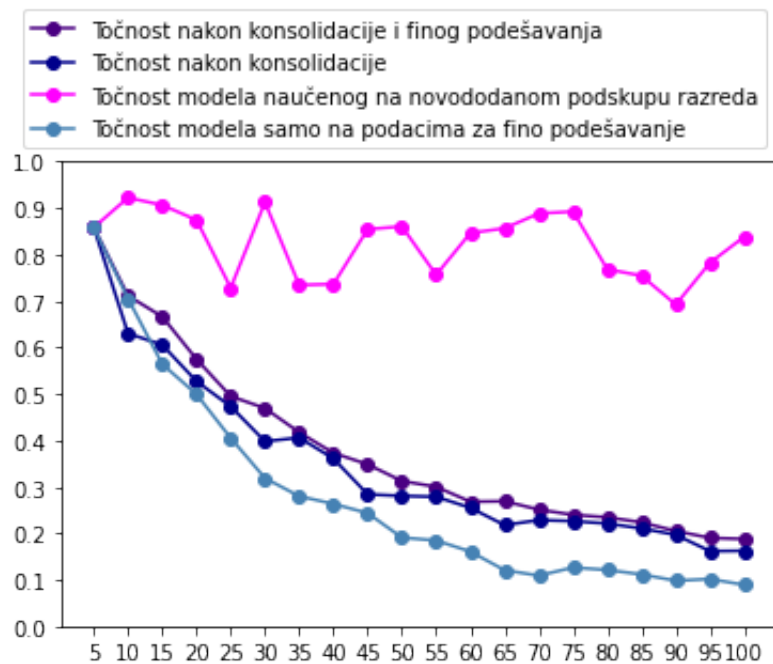
## 6.4. Konfiguracija 5/5

Kao što možemo iščitati iz tablice 6.1 konfiguracija 5/5 najteža je eksperimentalna postavka za inkrementalno učenje. Takvi rezultati su očekivani te su u skladu s trenutnim stanjem tehnologije u području inkrementalnog učenja uz dodavanje novih razreda. Nadalje, iz slike 6.7 možemo vidjeti, što je i očekivano, kako i u ovoj konfiguraciji postoje inherentno teške kombinacije razreda za klasifikaciju te raspon točnosti iznosi 23 postotnih bodova. S druge strane, konsolidirani model i u ovoj je konfiguraciji robustan na oscilacije u točnosti.

**Tablica 6.6:** Eksperimentalne postavke za konfiguraciju 5/5.

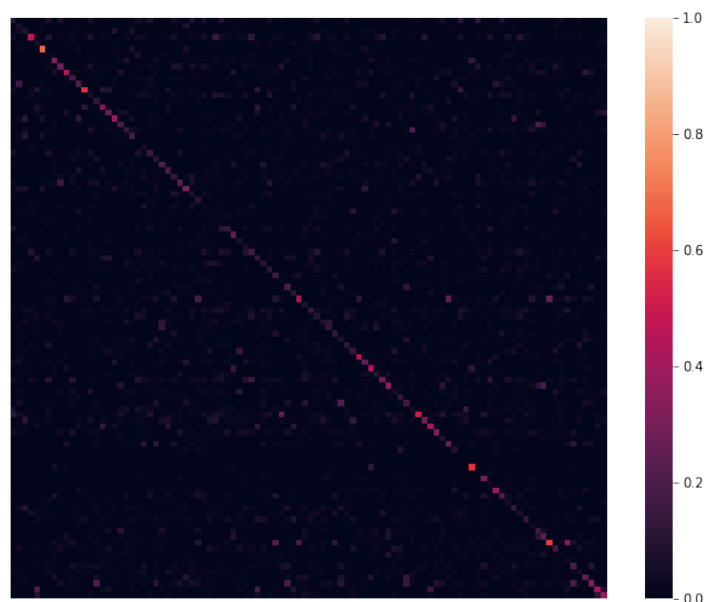
Naziv parametra	Vrijednost parametra
Korak učenja u fazi treniranja novog modela	0.01
Korak učenja u fazi konsolidacije	0.01
Faktor dijeljenja gubitka	10

Prosječno povećanje točnosti nakon finog podešavanja u ovoj konfiguraciji iznosi 3.23%. S obzirom na značajno smanjenje povećanja točnosti nakon faze finog podešavanja u konfiguracijama 10/10 i 5/5 opravdano je pitanje korisnosti ove faze u njihovom slučaju. Doprinos u odnosu na učenje s memorijskim budžetom i dalje je prisutan te je inkrementalno učenje za sve promatrane konfiguracije pokazalo svoju korisnost.



**Slika 6.7:** Točnosti inkrementalnog treniranja modela na konfiguraciji 5/5. Na osi apscisa prikazani su koraci inkrementalnog učenja.

Za kraj, u skladu s prethodnim konfiguracijama slika 6.8 pokazuje kako konsolidacija dubokih modela i u ovom eksperimentu ne pokazuje pristranost prema određenim klasama. Takav zaključak u skladu je s opažanjima iznesenima u [20].



**Slika 6.8:** Matrica zabune za konfiguraciju 5/5. Svaki redak i pripadajući stupac označavaju jedan razred. Redci predstavljaju predviđanja modela, a stupci točne razrede.

## 6.5. Rasprava

Iz svih prije opisanih rezultata vidimo kako faza finog podešavanja povećava točnost modela dobivenog konsolidacijom. Ipak, za eksperimentalne postavke s puno koraka inkrementalnog učenja značaj finog podešavanja se smanjuje. Kako slike starih razreda u memorijskom budžetu ostaju iste te samo izbacujemo određeni broj slika u svakom inkrementalnom koraku, s velikim brojem inkrementalnih koraka modelu se smanjuje količina korisnih informacija koja mu je dostupna jer je već vidio veliki broj slika.

Kao jedna od najvećih snaga ove metode pokazala se nepristranost prema starim ili novim razredima. Ključna za tu nepristranost je normalizacija u jednadžbi 4.3 i to što konsolidirani model nije inicijaliziran iz starog ili novog modela već nasumično. Iz rezultata ostalih algoritama inkrementalnog učenja koji su dostupni u [2] i [12] vidimo kako postoje algoritmi koji postižu bolju konačnu točnost od konsolidacije dubokih modela. Unatoč tome, konsolidacija dubokih modela ima svoju primjenu u slučajevima u kojima nam je bitno nepostojanje pristranosti prema određenim razredima. Daljnji rad na konsolidaciji dubokih modela uključivao bi procjenu iste na asimetričnim konfiguracijama. Drugim riječima, na konfiguracijama  $M/N$  gdje vrijedi  $M \neq N$ . Takve konfiguracije predstavljaju još realnije situacije te bi procjena kvalitete na njima bila značajna za primjenu inkrementalnog učenja.

## 7. Zaključak

Ovaj rad proučio je inkrementalno učenje dubokih modela uz dodavanje novih razreda. Na početku rada napravljen je povijesni i teorijski pregled dubokog učenja i detaljnije su opisani najbitniji algoritmi i komponente istih. Poseban naglasak stavljen je na konvolucijske neuronske mreže i njihovu primjenu u području raspoznavanja slika i računalnog vida općenito.

Nakon toga definiran je problem inkrementalnog učenja uz dodavanje novih razreda te su podijeljeni pristupi inkrementalnom učenju. Za daljnje razmatranje i implementaciju izabrana je metoda konsolidacije dubokih modela [20]. Za potrebe konsolidacije dubokih modela implementiran je i testiran ResNet-32. Kao skup podataka za evaluaciju ResNeta-32 i konsolidacije dubokih modela korišten je skup podataka CIFAR-100 kao što je predloženo u [14]. Za neoznačene pomoćne podatke korišten je skup za testiranje skupa slika ImageNet32.

Rezultati su pokazali potencijal konsolidacije dubokih modela uz korištenje finog podešavanja. Prosječni dobitak u krajnjoj točnosti nakon korištenja finog podešavanja kroz sve eksperimentalne postavke iznosi 4.78%. Korisnost memorijskog budžeta oscilira ovisno o eksperimentalnoj postavci te ovaj rad može poslužiti kao smjernica za procjenu korisnosti memorijskog budžeta pri konsolidaciji dubokih modela. Daljnje poboljšanje ovog rada može se ostvariti provođenjem novih eksperimenata za sve navedene konfiguracije s različitim poretkom razreda kako bi se kvantificirao utjecaj težine nekih skupina razreda.

# LITERATURA

- [1] S Agatonovic-Kustrin i R Beresford. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, 22(5):717–727, 2000. URL <https://www.sciencedirect.com/science/article/pii/S0731708599002721>.
- [2] Eden Belouadah, A. Popescu, i I. Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural networks : the official journal of the International Neural Network Society*, 135:38–54, 2021.
- [3] Bojana D. Bašić and Marko Čupić and Jan Šnajder. Umjetne neuronske mreže. [https://www.fer.unizg.hr/\\_download/repository/UmjetneNeuronskeMreze.pdf](https://www.fer.unizg.hr/_download/repository/UmjetneNeuronskeMreze.pdf), 2008.
- [4] P. Chrabaszcz, I. Loshchilov, i F. Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *ArXiv*, abs/1707.08819, 2017.
- [5] I. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, i Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *CoRR*, abs/1312.6211, 2014.
- [6] Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. Manning Publications Co, 2020. <http://www.deeplearningbook.org>.
- [7] Kaiming He, X. Zhang, Shaoqing Ren, i Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, stranice 770–778, 2016.
- [8] Geoffrey E. Hinton, Oriol Vinyals, i J. Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

- [9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [10] Johannes Lederer. Activation functions in artificial neural networks: A systematic overview. *ArXiv*, abs/2101.09957, 2021.
- [11] Zhizhong Li i Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2018.
- [12] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, i Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. 2020.
- [13] Michael McCloskey i Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. svezak 24 od *Psychology of Learning and Motivation*, stranice 109–165. Academic Press, 1989. URL <https://www.sciencedirect.com/science/article/pii/S0079742108605368>.
- [14] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, i Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, stranice 5533–5542, 2017.
- [15] Sebastian Ruder. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747, 2016.
- [16] Sinam Ajitkumar Singh, Takhellambam Gautam Meitei, i Swanirbhar Majumder. 6 - short pcg classification based on deep learning. U *Deep Learning Techniques for Biomedical and Health Informatics*, stranice 141–164. Academic Press, 2020. URL <https://www.sciencedirect.com/science/article/pii/B9780128190616000069>.
- [17] Stanford University. History: The 1940's to the 1970's. <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html>.
- [18] Niko Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, Michael Milford, i P. Corke. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37:405 – 420, 2018.

- [19] Thomas Wood. Backpropagation. <https://deeptai.org/machine-learning-glossary-and-terms/backpropagation>.
- [20] Junting Zhang, J. Zhang, S. Ghosh, Dawei Li, S. Tasci, Larry Heck, H. Zhang, i C.-C. Jay Kuo. Class-incremental learning via deep model consolidation. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, stranice 1120–1129, 2020.



## **Inkrementalno učenje modela za raspoznavanje slike uz dodavanje novih razreda**

### **Sažetak**

Inkrementalno učenje uz dodavanje novih razreda jedna je od grana dubokog učenja. Nadalje, posebno je zanimljiva primjena inkrementalnog učenja uz dodavanje novih razreda na problemu raspoznavanja slika. Inkrementalno učenje uz dodavanje novih razreda izrazito je korisno u sustavima za rad u stvarnom vremenu kao što su, na primjer, roboti. Jedna od novijih metoda u tom području je konsolidacija dubokih modela i ona je detaljnije proučena i implementirana u ovom radu. Implementacija metode testirana je na dobro poznatom skupu podataka CIFAR-100 za različite konfiguracije problema.

**Ključne riječi:** inkrementalno učenje uz dodavanje novih razreda, računalni vid, duboko učenje, konsolidacija dubokih modela, raspoznavanje slika

## **Class-incremental learning for image recognition**

### **Abstract**

Class-incremental learning is one of the branches of deep learning. Furthermore, the application of class-incremental learning to the problem of image recognition is particularly interesting. Class-incremental learning is very useful in real-time systems such as, for example, robots. One of the newer methods in this area is the deep model consolidation, and it has been studied in more detail and implemented in this paper. The implementation was tested on the well-known data set CIFAR-100 for different problem configurations.

**Keywords:** class-incremental learning, computer vision, deep learning, deep model consolidation, image recognition